SPATIO-TEMPORAL GRAPH NEURAL NETWORKS FOR MULTI-SENSOR FUSION IN FAULT DIAGNOSIS

Project Submitted in Partial Fulfilment of the Requirements for the Degree of

BACHELOR OF TECHNOLOGY
IN
INDUSTRIAL ENGINEERING

By

Kadam Sanika Narayanrao (22IM30036)

Under the Guidance of

Dr. Jhareswar Maiti
Professor

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
KHARAGPUR – 721302, INDIA
Autumn 2025-26 Session

DECLARATION

I hereby certify that

a. the work contained in this report is original and has been done by me under the guidance of my supervisor.
b. the work has not been submitted to any other Institute for any degree or diploma.
c. I have followed the guidelines provided by the Institute in preparing the report.
d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date:

Kharagpur

Kadam Sanika Narayanrao
22IM30036,
Department of Industrial and Systems Engineering,
Indian Institute of Technology,
Kharagpur, India.

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
KHARAGPUR – 721302, INDIA

CERTIFICATE

I hereby certify that the project report entitled, "**Spatio-Temporal Graph Neural Networks for Multi-Sensor Fusion in Fault Diagnosis**", submitted by **Kadam Sanika Narayanrao (22IM30036)** to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Industrial Engineering is a record of bona fide work carried out by her under my supervision and guidance during Autumn Semester, 2025-26.

Date:
Kharagpur

Dr. Jhareswar Maiti
Professor,
Department of Industrial and Systems Engineering,
Indian Institute of Technology,
Kharagpur, India.

Abstract

Name of the Student: **Kadam Sanika Narayanrao** Roll No: **22IM30036**
Degree for which submitted: **Bachelor of Technology**
Department: **Industrial and Systems Engineering**
Thesis Title: **Spatio-Temporal GNNs for Multi Sensor Fusion in Fault Diagnosis**
Thesis Supervisor: **Prof. Jhareswar Maiti**
Month and Year of Thesis Submission: **November, 2025**

The reliability of rotating machinery is essential in modern industry. Early detection of bearing faults is vital to prevent severe failures and reduce economic losses. While multi-sensor systems provide a better view of machine health compared to single-sensor approaches, traditional data fusion techniques often do not effectively model the physical and spatial relationships between sensors. This results in the loss of important diagnostic information. This project tackles this issue by introducing a new fault diagnosis framework that uses Spatio-Temporal Graph Neural Networks (ST-GNNs) for effective multi-sensor fusion.

This research redefines the multi-sensor fusion problem as a graph-level classification task. We suggest building a graph that illustrates the physical structure of the monitored system, with sensors acting as nodes and their physical connections as edges. The main part of the proposed method is a hybrid ST-GNN design. It combines a Graph Neural Network (GNN) to capture spatial dependencies and a Recurrent Neural Network (RNN) to model changes over time.

This research shows that while a standard 1D-CNN benchmark reaches high accuracy (97.82%) when trained on mixed-load data, the ST-GNN's success relies heavily on its input representation. A GNN using raw signals only achieved 78.99% accuracy on unseen loads, but its robustness increased to 94.90% when given multi-domain node features and correlation-based edge weights. This confirms that clearly modelling spatio-temporal relationships is a highly effective approach, as long as the nodes are represented by meaningful, load-invariant features.

Keywords: Fault Diagnosis, Multi-Sensor Fusion, Graph Neural Networks (GNNs), Spatio-Temporal Modelling, Predictive Maintenance, Deep Learning, CWRU Dataset.

Contents

List of Tables

List of Figures

Abbreviations

CNN: Convolutional Neural Network
CWRU: Case Western Reserve University
DE: Drive End
DTW: Dynamic Time Warping
EDM: Electro-Discharge Machining
FE: Fan End
FFT: Fast Fourier Transform
GAT: Graph Attention Network
GCN: Graph Convolutional Network
GNN: Graph Neural Network
GRU: Gated Recurrent Unit
HP: Horsepower
KNN: K-Nearest Neighbours
LSTM: Long Short-Term Memory
MLP: Multi-Layer Perceptron
MSIF: Multi-Sensor Information Fusion
PdM: Predictive Maintenance
REB: Rolling Element Bearing
ReLU: Rectified Linear Unit
RNN: Recurrent Neural Network
RMS: Root Mean Square
ST-GNN: Spatio-Temporal Graph Neural Network
SVM: Support Vector Machine

Chapter 1

Introduction

Bearings are among the most fundamental components of all rotating machines in an industry, transferring loads and reducing rotational friction during power transmission [2]. The health state of these bearings directly affects the precision and stability of machinery, making their timely maintenance essential for preventing breakdowns and ensuring reliability at a reduced cost. Failure of a Rolling Element Bearing (REB) is one of the most frequent problems in industrial settings, typically occurring in the rolling element, inner race, or outer race [2]. Detecting these faults at an early stage is crucial to prevent potential threats to assets and personnel, a practice broadly classified as predictive maintenance (PdM) [2]. Among various condition monitoring tools, vibration analysis is the most predominant technique, where data from sensors like accelerometers are analysed using machine learning to detect anomalies [2].

While vibration analysis is powerful, relying on a single sensor provides a localized and often incomplete view of a system's health, which can lead to misdiagnosis [3]. To address this, the paradigm has shifted towards Multi-Sensor Information Fusion (MSIF), which integrates data from multiple sensors to provide a more comprehensive and reliable assessment [4]. However, a fundamental challenge persists in how this data is fused. Traditional methods often treat sensor inputs as an unordered collection of data streams, overlooking the crucial information embedded in the physical layout and functional relationships between the sensors [3]. Recent advancements with Graph Neural Networks (GNNs) have begun to address this, but many state-of-the-art models construct complex, abstract graphs based on data similarity rather than the physical topology of the system [5]. This leads to a critical research gap: the lack of investigation into models that learn from a simple, physically-grounded representation of the sensor network. Furthermore, a significant challenge for all data-driven models is their poor generalization when faced with varying operating conditions, such as changes in speed or load [7].

This project directly addresses these gaps by investigating the following research question: **Can a Spatio-Temporal Graph Neural Network (ST-GNN), operating on a simple, physically-grounded graph, provide a more robust and generalizable fault diagnosis compared to methods that either ignore sensor topology or rely on complex, data-driven graph structures?** This research is significant as it aims to advance the field by testing a more intuitive and physically representative modelling approach, with the practical goal of creating more reliable and adaptable diagnostic tools for real-world industrial applications. To achieve this, we will develop an end-to-end ST-GNN framework that learns directly from raw multi-sensor time-series data. The model will be validated on the CWRU benchmark dataset, with a specific focus on evaluating its generalization performance across different operating loads. The expected contribution is a demonstration that explicitly modelling the physical propagation of fault signatures leads to a more robust diagnostic system.

This report is structured as follows: Chapter 2 presents a comprehensive literature review. Chapter 3 details the proposed methodology and model architectures. Chapter 4 presents the experimental results and discussion. Finally, Chapter 5 concludes the report, summarizing the key findings and outlining future research.

Chapter 2

Literature Review

This chapter provides a review of the existing literature relevant to multi-sensor fault diagnosis. It is organized into three main themes. First, it discusses the principles and limitations of traditional multi-sensor information fusion. Second, it introduces the foundational concepts of Graph Neural Networks (GNNs) and their variants. Finally, it surveys the state-of-the-art application of Spatio-Temporal GNNs in fault diagnosis, which allows for the precise identification of the research gaps and objectives of this project.

2.1 Multi-Sensor Information Fusion in Fault Diagnosis

The use of multi-sensor systems is a well-established strategy to overcome the limitations of single-sensor monitoring [2]. A single data stream can be incomplete or ambiguous, whereas fusing information from multiple sensors provides complementary and redundant data, leading to higher accuracy and reliability [4]. The fusion process is typically categorized into three levels:

- **Data-Level Fusion:** This involves the direct combination of raw sensor signals. While it theoretically preserves the maximum amount of information, it is highly sensitive to noise and requires the sensors to be synchronized and commensurate, which is often impractical in industrial settings [3].

- **Feature-Level Fusion:** This is the most common approach, where relevant features are extracted from each sensor signal and then concatenated into a single feature vector for classification. This reduces data dimensionality but risks losing critical information (e.g., phase relationships) during the feature extraction process. The performance of models like Support Vector Machines (SVMs) is heavily dependent on the quality of these hand-crafted features [2].

- **Decision-Level Fusion:** In this approach, each sensor has an independent classifier, and their final decisions are aggregated using rules like weighted voting. This method is robust to individual sensor failure but discards a significant amount of correlation information present at the lower levels [4].

A common thread across these traditional levels is their failure to explicitly incorporate the physical or functional topology of the sensor network. The fusion operation typically treats the inputs as an unordered set, forcing practitioners to manually engineer spatial context into the features—a brittle and expertise-intensive process. This conceptual gap highlights the need for a new class of models designed to operate natively on relational data.

2.2 Graph Neural Networks for Time Series Analysis

Graph Neural Networks (GNNs) have emerged as a powerful deep learning paradigm for analysing data with an underlying graph structure. Unlike traditional neural networks that operate on grid-like data (e.g., images) or sequences (e.g., text), GNNs can learn from the complex, non-Euclidean relationships between entities. The core mechanism of a GNN is message passing, where nodes iteratively aggregate information from their neighbours and update their own feature representations. This allows the model to learn features that capture both the attributes of the nodes and the structure of their local neighbourhood.

Several GNN variants have been developed, with two of the most foundational `being:

- Graph Convolutional Networks (GCNs): GCNs perform a localized spectral filtering operation, which can be interpreted as a learnable, weighted average of a node's features and its neighbours' features. They are computationally efficient and have been widely used as a foundational GNN layer [1].
- Graph Attention Networks (GATs): Unlike GCNs which use fixed weights defined by the graph's adjacency matrix, GATs use a self-attention mechanism to learn the relative importance of different neighbours for each node. This allows for more flexible and powerful spatial feature aggregation [1].

The ability of GNNs to explicitly model inter-variable relationships make them a natural fit for analysing multivariate time series from sensor networks, where the variables (sensors) have known or learnable relationships.

2.3 Spatio-Temporal GNNs in Fault Diagnosis

The application of GNNs to dynamic systems like machinery monitoring has led to the development of Spatio-Temporal Graph Neural Networks (ST-GNNs). These models are designed to simultaneously learn from both the spatial structure of the sensor network and the temporal evolution of the signals [1]. A common and effective architecture is a hybrid model that combines a spatial GNN module with a temporal RNN module (like LSTM or GRU) [1].

A review of recent literature shows a strong trend towards using ST-GNNs for fault diagnosis, but with different philosophies regarding graph construction:

- In [2] a classical SVM approach is used, which achieves high accuracy (97.36%) on the CWRU dataset but relies entirely on manual feature engineering from a single sensor signal, highlighting the limitations that multi-sensor GNNs aim to overcome.

- [4] uses a GNN for multi-sensor fusion but employs a spatial-only model. It focuses on a static snapshot of the system and uses a complex autoencoder for feature extraction and a partially learned graph structure.

- [8] presents a powerful model that achieves near-perfect accuracy. However, it is a single-sensor technique that creates an abstract graph where nodes are signal segments connected by frequency-domain similarity, rather than modelling a physical sensor network.

- [5] and [6] are highly relevant as they use ST-GNNs. However, their novelty lies in constructing complex, data-driven graphs based on similarity metrics like Dynamic Time Warping (DTW) or K-Nearest Neighbours (KNN).

These state-of-the-art methods invest significant complexity in learning an optimal graph topology from the data itself. This leaves an open question about the effectiveness of a simpler, more intuitive approach.

2.4. Research Gaps

Based on the comprehensive literature review, the following research gaps have been identified:

1. **Lack of Physical Grounding:** While ST-GNNs are used for fault diagnosis, many depend on abstract, data-driven graph construction methods. There is little research on how these models perform on a simple graph that directly shows the physical layout of the sensor network.
2. **Modelling of Fault Propagation:** Using a physically grounded graph offers a natural way to model how fault-induced vibrations move through a machine's structure. This effect is not clearly captured by abstract similarity-based graphs.
3. **Generalization Under Varying Conditions:** A persistent challenge in the field is the poor generalization of models to operating conditions not seen during training [7]. While some studies tackle this with complex domain adaptation methods, the potential strength of a physically grounded ST-GNN has not been fully examined as a solution.

2.5. Research Objectives

This project aims to address the identified research gaps by pursuing the following objectives:

1. To develop a multi-sensor fusion framework for bearing fault diagnosis using a Spatio-Temporal Graph Neural Network, and to benchmark its performance against a standard 1D Convolutional Neural Network.
2. To investigate the efficacy of using a physically-grounded graph topology to model inter-sensor relationships, testing the hypothesis that this provides a powerful yet simple framework for learning fault dynamics.
3. To explicitly test the impact of node feature representation on GNN performance by comparing models trained on raw signals versus models trained on handcrafted, multi-domain features (time and frequency).
4. To evaluate the proposed models' robustness and generalization capabilities using a strict mixed-load training strategy: training on 0, 1, and 2 HP load data and testing on completely unseen 3 HP load data from the CWRU benchmark dataset [10]

Chapter 3

Methodology

This chapter details the data, preprocessing steps, and model architectures used to investigate the research objectives. The methodology is designed as a phased experiment to systematically test different strategies for improving diagnostic accuracy and robustness.

3.1 Dataset Description

This project uses the industry-standard Case Western Reserve University (CWRU) bearing dataset [10]. This dataset is widely used for benchmarking fault diagnosis models.

- **Experimental Setup**: The data was collected from a 2 HP induction motor test rig. Faults of varying diameters (0.007 to 0.021 inches) were seeded onto the bearings using electro-discharge machining (EDM).

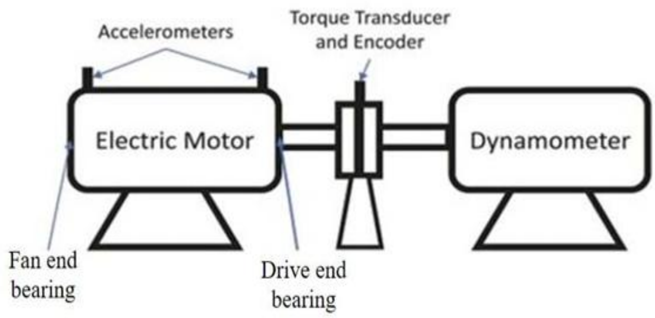

Fig. 1 Schematic of the experimental setup for data collection [2]

- **Sensor Data:** Vibration data was collected using accelerometers placed at the Drive End (DE) and Fan End (FE) of the motor housing. Data was sampled at 12 kHz.

- **Operating Conditions:** The motor was run under four different load conditions: 0 HP, 1 HP, 2 HP, and 3 HP.

- **Fault Classes:** This project focuses on classifying four distinct health states: Normal (Healthy), Ball Fault, Inner Race Fault, and Outer Race Fault

The use of both DE and FE sensor data is central to the multi-sensor fusion approach of this project.

3.2 Data Preprocessing and Graph Formulation

A robust data pipeline is critical for deep learning models. The following steps are applied to the raw 12 kHz vibration data.

3.2.1 Signal Segmentation

The continuous time-series signals from both DE and FE sensors are segmented into smaller, overlapping windows. Each window becomes a single data sample.

- **Window Size:** 1024 data points. This size is large enough to capture one or more full shaft rotations and associated fault frequencies.
- **Overlap:** 512 data points (50% overlap). This technique of using a sliding window significantly augments the number of available training samples.

3.2.2 Data Splitting and Scaling

To test model robustness and generalisation, a strict mixed-load training strategy is employed, as this is a known pitfall for many diagnostic models [7].

1. **Dataset-Level Split:** The raw data files are first divided into three sets based on operating load:

- **Training/Validation Set:** All files from 0 HP, 1 HP, and 2 HP loads.
- **Unseen Test Set:** All files from the 3 HP load.

2. **File-Based Split:** The (0, 1, 2 HP) set is further split by file into a 90% training set and a 10% validation set.
3. **Segmentation:** Only after the files are split are the signals within them segmented into windows.
4. **Scaling:** A *StandardScaler* is *fit* only on the training segments (from the 0, 1, 2 HP files). This fitted scaler is then used to *transform* all datasets (training, validation, and the unseen 3 HP test set). This ensures the test set remains completely unseen and is processed identically to the training data.

3.2.3 Graph Formulation

The core hypothesis of this project is tested by formulating the multi-sensor system as a simple, physically-grounded graph.

- **Nodes:** The graph consists of two nodes, representing the two physical sensors: V_{DE} (Drive End) and V_{FE} (Fan End).
- **Edges:** A single undirected edge (V_{DE}, V_{FE}) is created to represent the physical connection between the sensors via the motor housing.

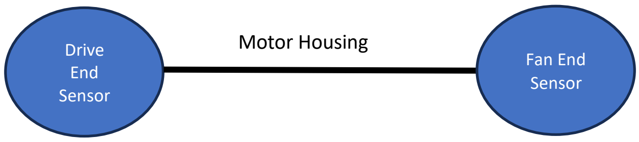

Fig. 2 Physically-Grounded Graph Structure

- **Node Features:** For each data sample (i.e., each time window), the node features are the 1024-point segmented vibration signals from the corresponding sensor

This results in a dataset of graph-structured samples, where each graph has 2 nodes and 2 edges (one in each direction). The features for these nodes vary by experiment.

3.2.4 Edge Weight Formulation

To test the hypothesis that providing physical context improves model robustness, the ST-GNN v3 model uses a weighted graph. The edge weight is formulated to represent the strength of the physical connection between the sensors.

- **Metric:** The weight for the edge (V_{DE}, V_{FE}) is calculated as the absolute Pearson correlation coefficient between the *full* raw signals from the Drive-End sensor (X) and the Fan-End sensor (Y) for each data file.
- **Formula:** The Pearson correlation coefficient, *r*, is calculated as:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

Where:

n is the number of data points in the signal.

x and y are the individual signal values for the DE and FE sensors, respectively.

- **Implementation:** This calculation is performed once per file, providing a single, dynamic edge weight (|r|) for all segments derived from that file. This weight is then passed to the Graph Attention (GAT) layer during training and evaluation.

3.3 Feature Engineering

While deep learning models can learn from raw signals, their performance can often be improved by providing them with a more concise and informative feature representation. For the advanced models (ST-GNN v2 and v3), we replace the raw 1024-point signal with an 11-dimensional feature vector extracted from each segment.
This "multi-domain" vector provides statistical information from the time domain and energy information from the frequency domain. The extracted features are detailed in Table 1, based on the approach in [2].

Table 1: Details of the extracted features

Features	Formula	Domain	Description
Mean	$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$	Time	Average signal value
RMS	$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}$	Time	Root Mean Square (Signal Energy)
Skewness	$x_{skew} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^3}{(N-1)\sigma^3}$	Time	Measure of signal asymmetry
Kurtosis	$x_{kurt} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})^4}{(N-1)\sigma^4}$	Time	Measure of signal "peakedness"
P2P	$x_{p2p} = \max(x_i) - \min(x_i)$	Time	Peak-to-Peak (Amplitude)
Crest Factor	$x_{crest} = \frac{\max(\lvert x_i \rvert)}{x_{rms}}$	Time	Ratio of peak to RMS (Impact)
FFT Band (i=1-5)	$E_i = \frac{1}{N_b}\sum_{k=S_i}^{e_i} \lvert X_k \rvert$	Frequency	Mean energy distribution across 5 frequency bands (0-6.0 kHz)

3.4 Model Architectures

Four different model architectures are defined to test the hypotheses regarding feature representation and spatio-temporal modelling. All models are trained on the mixed-load data.

3.4.1 Model 1: 1D-CNN Benchmark

This model provides a solid starting point. It uses a 1D Convolutional Neural Network (CNN) that takes the raw signals from both sensors as input. The input shape is [Batch_Size, 2, 1024], where 2 stands for the two sensor channels (DE and FE). The architecture includes three 1D convolutional blocks: Conv1D, BatchNorm, ReLU, and MaxPool. This is followed by two fully-connected layers for classification. The model merges the sensor data at the first convolutional layer.

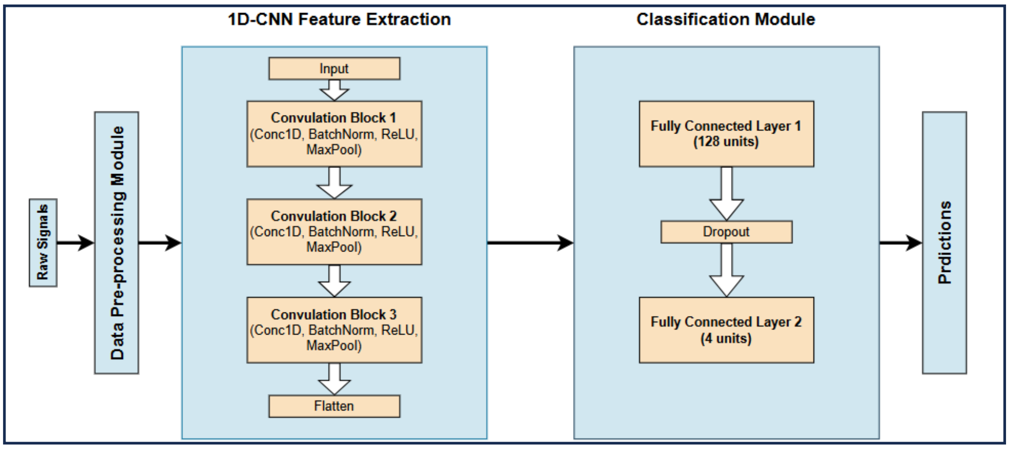

Fig 3 1D-CNN Architecture

3.4.2 Model 2: ST-GNN v1

This is the first spatio-temporal model. It processes the 1024-point raw signal from each node independently. An LSTM layer captures temporal features. The final hidden state of the LSTM from each node, which is a feature vector, is then sent to a Graph Attention (GAT) layer. The GAT layer combines information spatially by learning the attention weights between the DE and FE nodes. This happens before a final graph-level pooling and classification step.

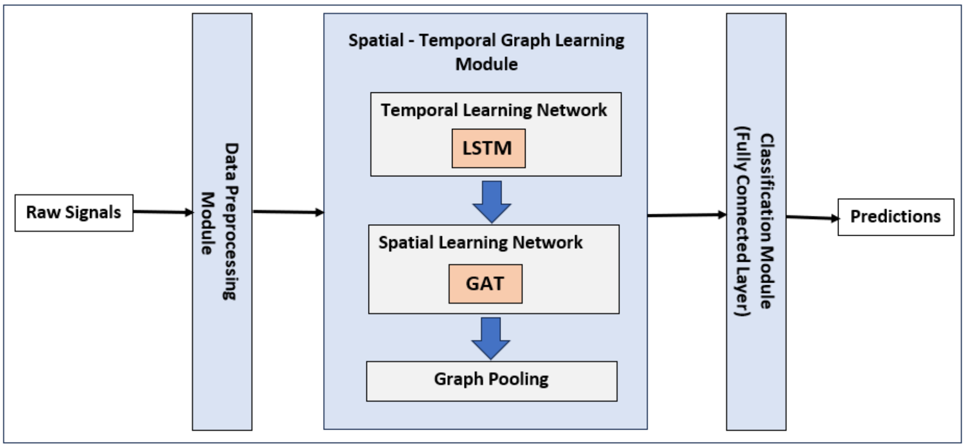

Fig 4 ST-GNN v1 Architecture

3.4.3 Model 3: ST-GNN v2

This model has the *exact same architecture* (Fig. 4) as ST-GNN v1, but its input features are different. One more step of feature engineering is added in data preprocessing module. Instead of feeding the 1024-point raw signal into the LSTM, it is fed the 11-dimensional handcrafted feature vector (from Table 1). This tests the hypothesis that better node features will improve the performance of the spatio-temporal architecture.

3.4.4 Model 4: ST-GNN v3

This model extends ST-GNN v2 by introducing a physically meaningful edge weight. The GAT layer is modified to accept an edge_weight attribute. This weight is calculated for each data sample as the absolute Pearson correlation coefficient between the raw DE and FE signals for that file. This tests the hypothesis that explicitly telling the GNN how strongly the sensors are correlated improves its ability to fuse their information.

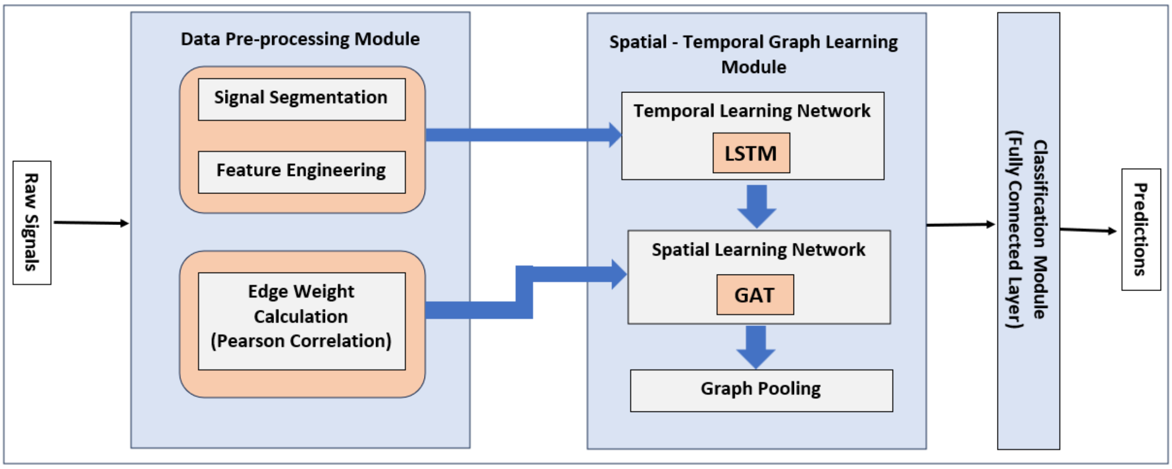

Fig 5 ST-GNN v3 Architecture

3.5 Hyperparameter Tuning

To ensure a fair comparison, all four models are optimized using the **Optuna** hyperparameter tuning library. An objective function is created for each model, which Optuna attempts to maximize over 25 trials.

- **Objective:** Maximize the validation accuracy on the *val_loader*.
- **Search Space:**
    - *learning_rate*: Log-uniform distribution between 1e-4 and 1e-2.
    - *dropout_rate*: Uniform distribution between 0.1 and 0.5.
    - *rnn_hidden_size*: Categorical choice [16, 32, 64] (for GNNs).
    - *gnn_hidden_size*: Categorical choice [32, 64, 128, 256] (for GNNs).
- **Training:** Each trial is trained for 20 epochs with an Adam optimizer.

3.6 Performance Evaluation

After finding the best hyperparameters, each model is re-trained from scratch for 20 epochs on its respective training data. The final trained models are then evaluated on two separate test sets:

- **Same-Distribution Test Set:** This includes data from the 0, 1, and 2 HP loads that was never used in training or validation. This tests how well the model learned from the training distribution.
- **Unseen-HP Test Set:** This includes data from the 3 HP load. This is the main measure of robustness and generalization since the model has never seen data from this operating condition.

The model's classification performance is quantified using standard metrics derived from the confusion matrix, which compares actual labels to predicted labels. These metrics—True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)—are used to calculate accuracy, precision, recall, and f1-score.

Table 2: Evaluation Metrices

Evaluation Metric	Formula	Evaluation Metric	Formula
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Recall	$\frac{TP}{TP + FN}$
Precision	$\frac{TP}{TP + FP}$	F1-Score	$2 \times \frac{Precision \times Recall}{Precision + Recall}$

Chapter 4

Results and Discussion

This chapter presents the results of the experiments detailed in the methodology. It begins with a visual exploration of the data, followed by a summary of the optimal hyperparameters found, and concludes with a comprehensive performance evaluation of the four models, including their classification reports and confusion matrices.

4.1 Data Exploration

Visualizing the data is essential for understanding the classification challenge. Fig 6 shows the raw vibration signals from both the Drive-End (DE) and Fan-End (FE) sensors for each of the four health states, using data from the 3HP load. The Normal state displays a low-amplitude, uniform signal. In contrast, the Ball, Inner Race, and Outer Race faults each show distinct, high-amplitude impact patterns. This visual difference confirms the presence of fault signatures and emphasizes the difficulty of distinguishing between the three fault types, especially across different sensors.

Figure 7 illustrates the sliding window segmentation process. The full raw signal (top) is divided into 1024-point windows (bottom) with a 50% overlap. Finally, Figure 8 demonstrates the feature engineering process. The table lists the 11 extracted features for the first 5 segments of a sample signal.

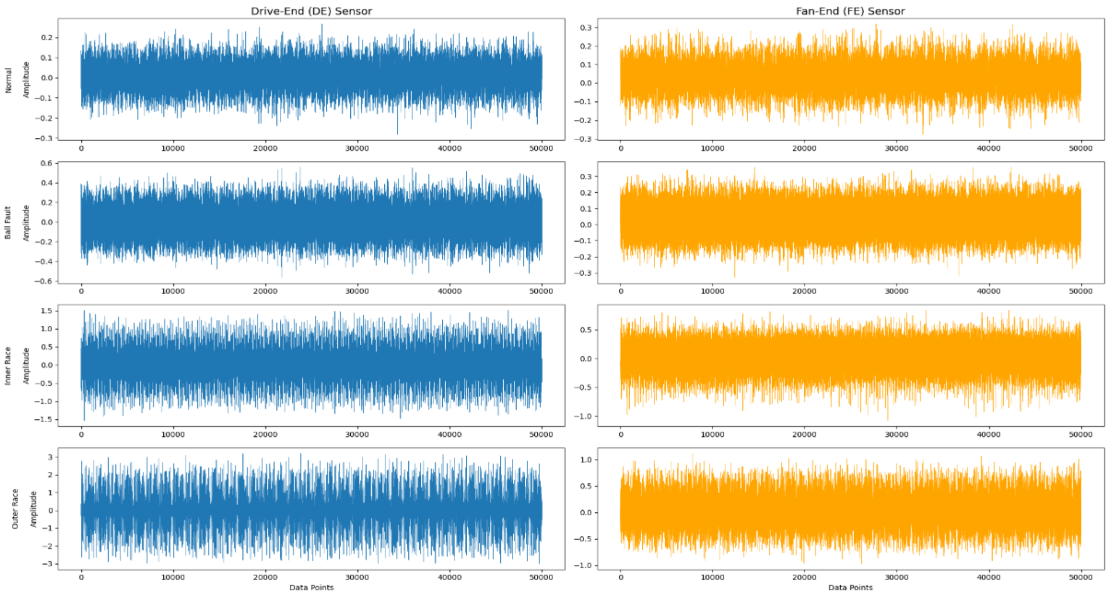

Fig 6: Raw Vibration Signals by Fault Type (3HP Load)

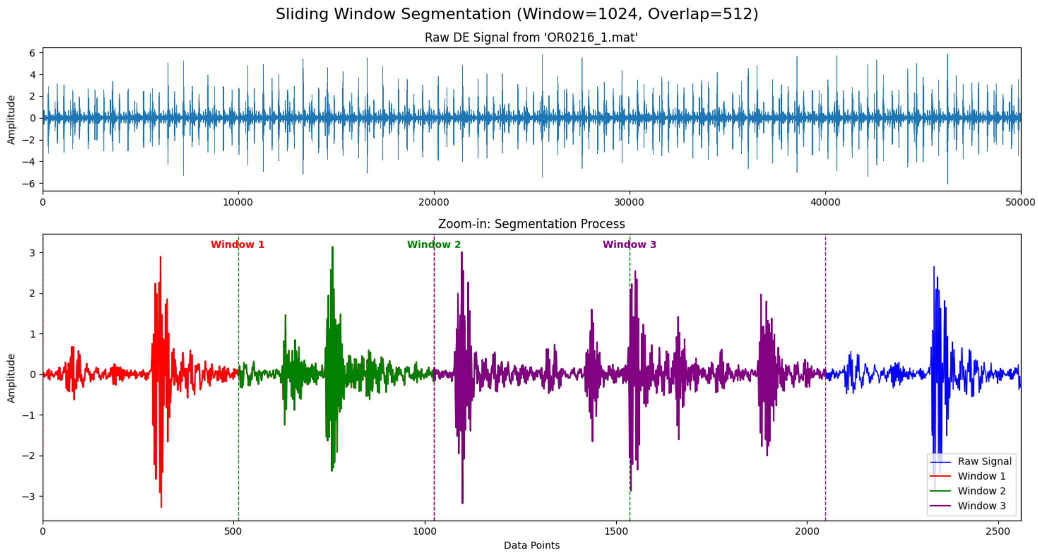

Fig 7: Sliding Window Segmentation Visualization

--- Feature Extraction Example (first 5 segments) ---

	Mean	RMS	Skewness	Kurtosis	P2P	CrestFactor	FFT_Band1	FFT_Band2	FFT_Band3	FFT_Band4	FFT_Band5
0	-0.019691	1.283501	-0.318030	15.160762	17.556013	6.995683	0.031251	0.025741	0.132309	0.030386	0.006413
1	-0.014755	1.311615	0.179305	12.662182	17.478103	6.773414	0.028717	0.026979	0.137282	0.032039	0.006015
2	-0.017203	1.436025	-0.130843	9.099644	17.286507	6.165006	0.032371	0.027308	0.147744	0.032881	0.006032
3	-0.022429	1.416402	-0.317094	9.239804	15.249620	5.695968	0.040871	0.027803	0.147549	0.033053	0.008627
4	-0.018809	1.285501	-0.117835	11.090155	14.846944	5.849218	0.031679	0.026401	0.137074	0.028769	0.005847

Fig. 8: Feature Extraction Example

4.2 Hyperparameter Tuning

All four models were tuned using the Optuna library over 25 trials, optimizing for validation accuracy on the mixed-load validation set (0, 1, 2 HP data). The final optimal hyperparameters used for training are presented in Table 3.

Table 3: Optimal Hyperparameters from Optuna Study

Model	dropout_rate	lr	rnn_hidden_size	gnn_hidden_size
1D-CNN	0.3952	0.004643	N/A	N/A
ST-GNN v1	0.3945	0.008843	64	64
ST-GNN v2	0.1147	0.002739	64	128
ST-GNN v3	0.2327	0.004341	32	128

4.3 Performance Evaluation

Using the best hyperparameters, each model was retrained for 20 epochs. The models were then evaluated on the "Same-Distribution" test set (0, 1, 2 HP loads) and the "Unseen-HP" test set (3 HP load) to assess their robustness.

Table 4 offers a comparison of the models, while Table 5 presents a detailed breakdown of performance metrics for the unseen 3HP test set. The confusion matrices in Figure 8 show the class-level performance on this unseen 3HP data.

Table 4: Final Experimental Summary Table

	Architecture	Input Type	Same-Distribution Accuracy (0, 1, 2 HP Test Set)	Unseen-HP Accuracy (Robustness)
1D-CNN	1D-CNN	Raw Signal	99.92 %	97.82 %
ST-GNN v1	LSTM→GAT	Raw Signal	77.36 %	78.99 %
ST-GNN v2	LSTM→GAT	11 Features	90.12 %	92.56 %
ST-GNN v3	LSTM→GAT	11 Features+ Weighted Edge	96.96 %	94.90 %

Table 5: Detailed Performance Summary (Unseen 3HP Test Set)

Model	**1D-CNN**			**ST-GNN v1**			**ST-GNN v2**			**ST-GNN v3**		
	F1-score	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**
Ball Fault	0.9549	0.9138	1.000	0.7402	0.5885	0.9972	0.8346	0.8773	0.7958	0.8891	0.9227	0.8577
Inner Race	0.9506	1.0000	0.9058	0.6033	0.9515	0.4416	0.9367	0.8991	0.9775	0.9560	0.9342	0.9789
Outer Race	1.0000	1.0000	1.0000	0.7080	0.7755	0.6512	0.9026	0.8994	0.9058	0.9341	0.9219	0.9466
Normal	1.0000	1.0000	1.0000	0.9989	0.9979	1.0000	0.9995	1.0000	0.9989	0.9984	1.0000	0.9968

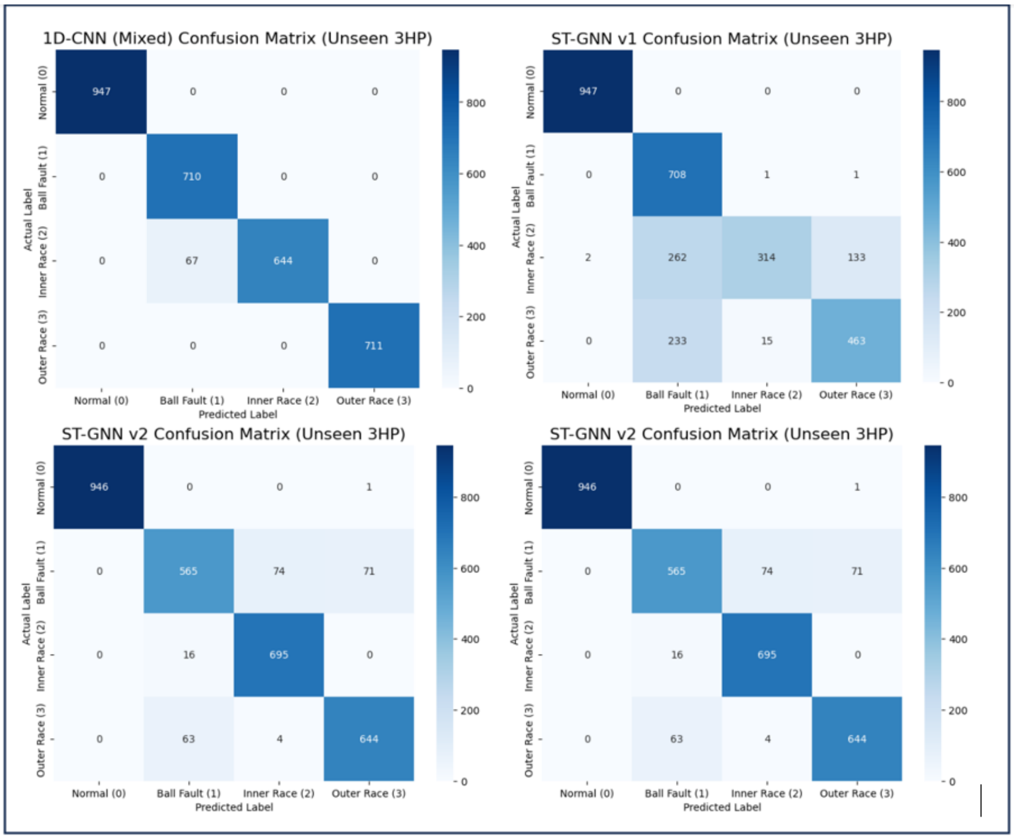

Fig 9: Confusion Matrices (Unseen 3HP)

4.4 Discussion

The results in the tables and confusion matrices provide clear answers to the project's main research questions.

1. **1D-CNN Benchmark**: The 1D-CNN (Table 5) set a high benchmark, achieving 97.82% accuracy on the unseen 3HP load. Its confusion matrix (Fig. 9) confirms this strong performance. It shows nearly perfect classification, with only minor confusion between Inner Race (2) and Ball Fault (1) faults.
2. **Impact of Node Features (ST-GNN v1 vs. v2)**: The ST-GNN v1 model, which used raw signals, had poor performance at 78.99% accuracy. Fig 9 highlights this failure, as it often misclassified Inner Race and Outer Race faults, confusing them with Ball Faults. In contrast, switching to the 11 handcrafted features allowed ST-GNN v2's accuracy to rise significantly to 92.56% (Table 4). Figure 8 shows a major improvement in identifying all fault types. This confirms that providing the GNN with useful, multi-domain node features is much more effective than having it learn from the long, raw time series directly.
3. **Impact of Physical Edge Weights (ST-GNN v2 vs. v3):** Adding the meaningful correlation-based edge weights further improved performance. ST-GNN v3 achieved 94.90% accuracy (Table 4). A comparison of confusion matrices (Fig. 9) demonstrates that this model is more balanced, particularly with better recall for Ball Faults. This supports the idea that explicitly modeling the strength of the physical sensor relationship enhances the GNN's ability to fuse information.

In summary, while the 1D-CNN was the strongest model overall, the experiments showed a clear performance improvement from v1 to v2 to v3. This confirms the core project ideas: GNN performance relies heavily on informative node features (v2 vs. v1), and explicitly modeling physical relationships (v3 vs. v2) provides a noticeable advantage.

Chapter 5

Conclusion and Future Scopes

5.1 Conclusion

This project successfully evaluated a Spatio-Temporal Graph Neural Network (ST-GNN) framework for multi-sensor fault diagnosis using the CWRU dataset. The 1D-CNN benchmark was the best model, achieving 97.82% accuracy. However, the performance of the ST-GNN framework showed a clear improvement, which supported the project's hypotheses.
The raw-signal ST-GNN v1 performed poorly with an accuracy of 78.99%. However, using 11 handcrafted features in ST-GNN v2 significantly raised the accuracy to 92.56%. Adding correlation-based edge weights in ST-GNN v3 further increased robustness to 94.90%. This confirms that GNNs can be a good alternative if they have informative node features and that modelling the physical structure explicitly is an effective way to combine strategies.

5.2 Future Scope

This project opens several paths for future research:

1. Explore more advanced architectures. Use Transformers for temporal modelling or different GNNs like GCN and GraphSAGE.
2. Implement dynamically learned edge weights with an attention mechanism instead of pre-calculated correlation.

3. Replace manual feature engineering with an automated feature extractor, like a 1D-CNN or Autoencoder, to learn optimal node features.
4. Apply this framework to more complex machinery. Work with larger, more intricate sensor graph structures.
5. Validate the framework's generalizability on other benchmark datasets.

References

[1] Corradini, F., Gerosa, F., Gori, M., Lucheroni, C., Piangerelli, M., & Zannotti, M. (2025). *A Systematic Literature Review of Spatio-Temporal Graph Neural Network Models for Time Series Forecasting and Classification*. arXiv preprint arXiv:2410.22377.

[2] Gupta, S., Maiti, J., & Kumar, A. (2023). An Optimized Machine Learning Technique for Fault Diagnosis of Roller Bearing in a Motor Drive System. In *2023 4th International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 616-622). IEEE.

[3] Liu, Z., & Li, Y. (2025). A Systematic Review of Multi-Sensor Information Fusion for Equipment Fault Diagnosis. *IEEE Transactions on Instrumentation and Measurement*.

[4] Jiang, G., et al. (2025). Heterogeneous Multi-Sensor Fusion for AC Motor Fault Diagnosis via Graph Neural Networks.
Electronics, 14(10), 2005.

[5] Singh, M. T., et al. (2024).
Spatial-Temporal Bearing Fault Detection Using Graph Attention Networks and LSTM. arXiv preprint arXiv:2410.11923.

[6] Xiao, Z., Cao, X., Hao, H., Liang, S., Liu, J., & Li, D. (2025). A Spatio-Temporal Joint Diagnosis Framework for Bearing Faults via Graph Convolution and Attention-Enhanced Bidirectional Gated Networks.
Sensors, 25(13), 3908.

[7] Peng, T., et al. (2025). Research on Bearing Fault Diagnosis Method for Varying Operating Conditions Based on Spatiotemporal Feature Fusion.
Sensors, 25(13), 3908.

[8] Zhou, K., et al. (2025). Rolling Bearing Fault Diagnosis via Temporal-Graph Convolutional Fusion Network (T-GCFN).
Sensors, 25(13), 3894.

[9] Jiang, G., Shen, K., Liu, X., et al. (2024). Prior Knowledge-Informed Graph Neural Network with Multi-Source Data-Weighted Fusion for Intelligent Bogie Fault Diagnosis.
Applied Sciences, 12(12), 838.

[10] Loparo, K. A. (n.d.).
Bearing Data Center. Case Western Reserve University. Retrieved from https://engineering.case.edu/bearingdatacenter

[11] Ma, J. (2022). Bearing fault diagnosis under class unbalanced data based on deep learning.
Academic Journal of Engineering and Technology Science, 5(3), 5-12.

[12] Zhang, Y., & Li, H. (2022). Rolling Bearing Fault Diagnosis Based on Graph Convolution Neural Network.
ResearchGate.